THE UNIVERSITY OF TEXAS AT ARLINGTON, TEXAS
DEPARTMENT OF ELECTRICAL ENGINEERING

EE 5322 - 002

INTELLIGENT CONTROL SYSTEMS

EXAM 1

by

SOUTRIK PRASAD MAITI

1001569883

Presented to

Dr. Frank Lewis

Oct 10, 2017

EE 5322 Intelligent Control- Exam 1
Fall 2017
1. This is a take home exam. YOU MUST WORK ALONE.
Any cheating or collusion will be severely punished.
2. To obtain full credit, show all your work. No partial credit will be given without the supporting
work.
3. Please sign this form and include it as the first page of your submitted exam.
....
...
Typed Name: ______Soutrik Maiti______________________________________
Pledge of honor:
"On my honor I have neither given nor received aid on this examination."
e-Signature: ____________Soutrik Maiti____________________________

EXAM 1

Mobile Robot Control and Potential Fields

Problem 1:

MATLAB Code –

```
% 13X13 grid
[x,y] = meshgrid(0:0.26:13);

% Gain of target and the two obstacles
k=2.89;
k1=4.96;
k2=5.7;

% Vector to the target (11,12)
r = sqrt((11-x).^2 + (12-y).^2);

% Attractive potential for the Target
Vt = k*r;

% Vector to the first Obstacle (3,4)
ro1 = sqrt((3-x).^2 + (4-y).^2);

% Repulsive Potential of the first Obstacle
Vro1 = k1./ro1;

%Vector to the second Obstacle (9,7)
ro2 = sqrt((9-x).^2 + (7-y).^2);

% Repulsive potential to the second Obstacle
Vro2 = k1./ro2;

% Total Potential
V = Vt+Vro1+Vro2;

% Plotting the Potentials

figure(1)
meshc(x,y,Vt)
title('Potential due to Target')
xlabel('X-axis')
ylabel('Y-axis')

figure(2)
meshc(x,y,Vro1)
title('Potential due to Obstacle 1')
xlabel('X-axis')
```

```
ylabel('Y-axis')

figure(3)
mesh(x,y,Vro2)
title('Potential due to Obstacle 2')
xlabel('X-axis')
ylabel('Y-axis')
figure(4)
meshc(x,y,V)
title('Potential Field')
xlabel('X-axis')
ylabel('Y-axis')
```

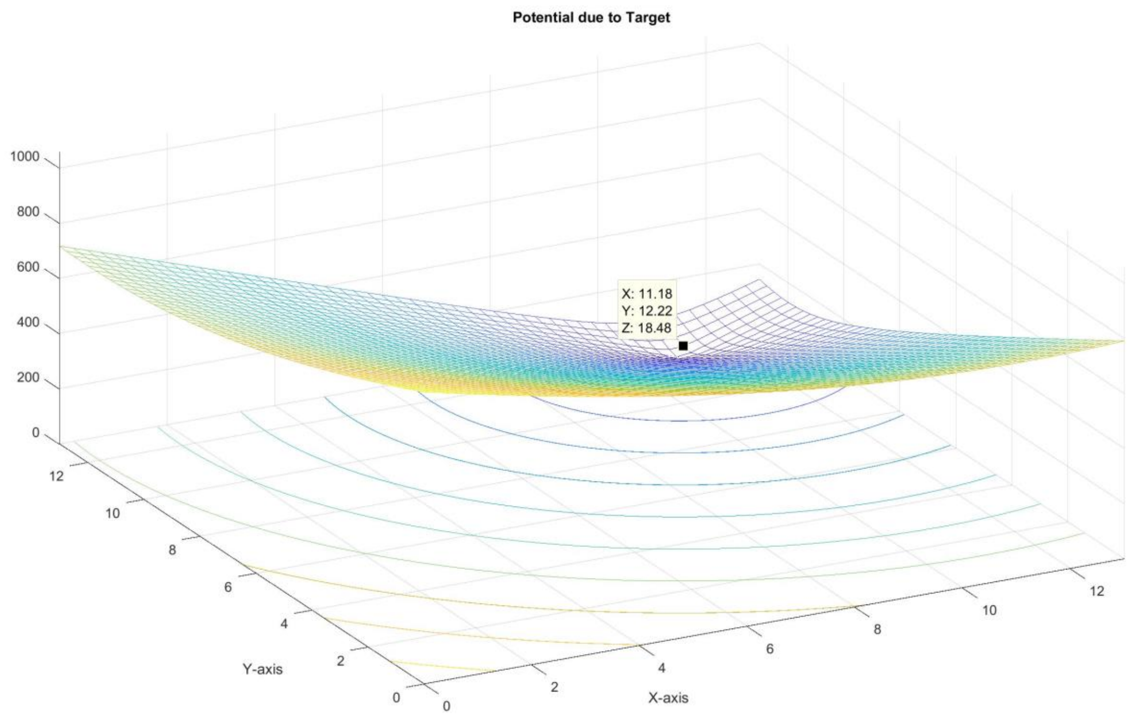


Fig E1- Potential due to target at (11,12)

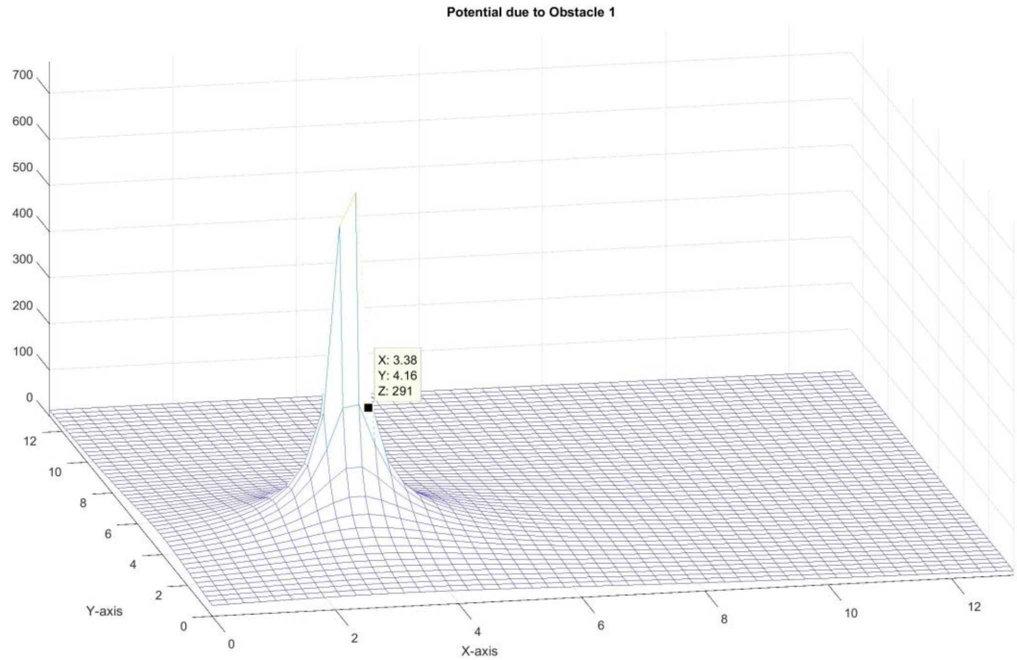


Fig E2- Potential due to obstacle at (3,4)

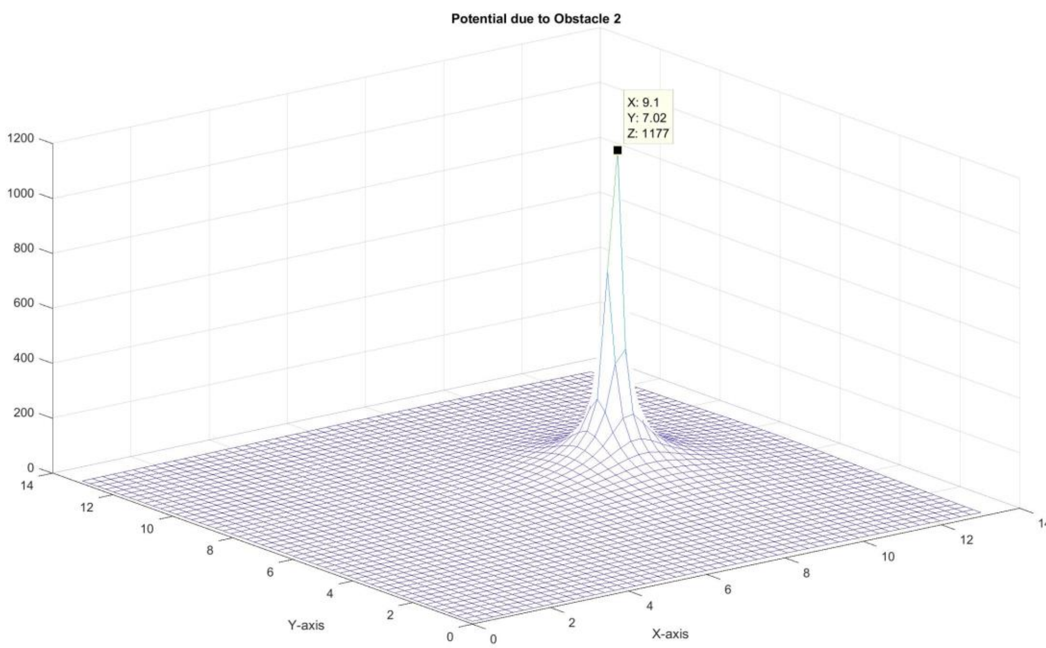


Fig E3- Potential due to obstacle at (9,7)

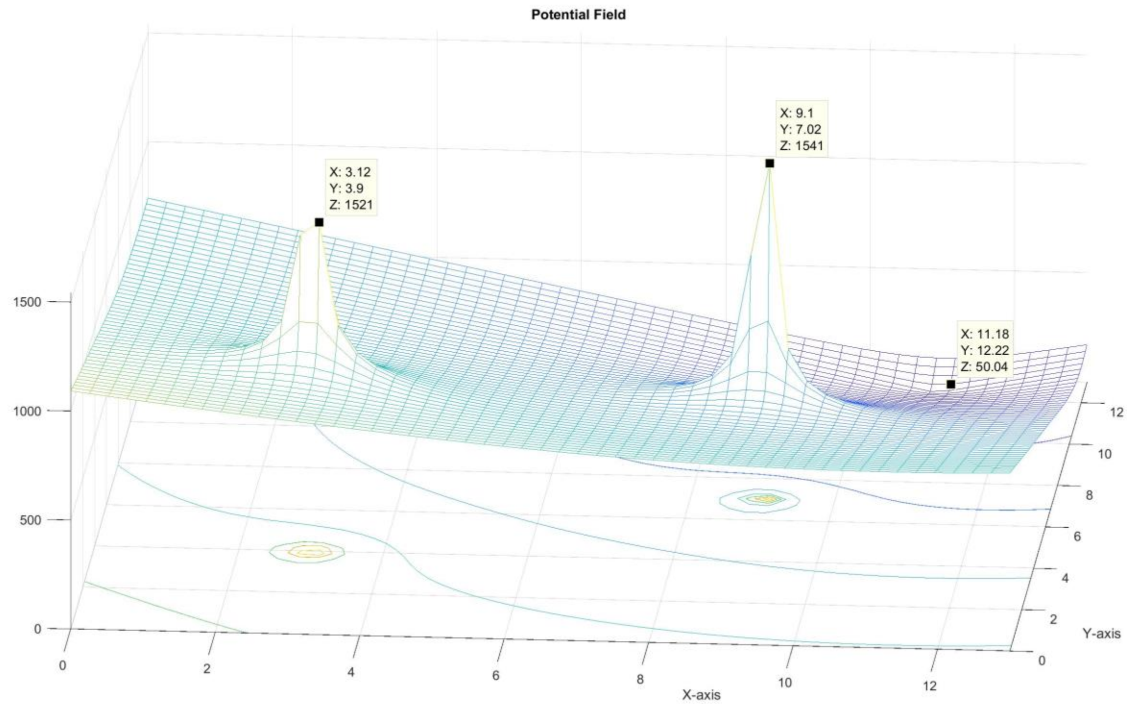


Fig E4- Total potential field

Problem 2 :

MATLAB Code –

```
% 13X13 grid
[x,y] = meshgrid(0:0.26:13);

% Gain of target and obstacles
k1 =2;
k2 =6;
k3 = 7;

% Vector to the target
r = sqrt((11-x).^2 + (12-y).^2);

% Attractive potential for the Target
Vg = k1*r;

% Vector to the first Obstacle (3,4)
r1 = sqrt((3-x).^2 + (4-y).^2);

% Repulsive Potential of the first Obstacle
```

```matlab
Vr1 = k2./r1;

%Vector to the second Obstacle (9,7)
r2 = sqrt((9-x).^2 + (7-y).^2);

% Repulsive potential to the second Obstacle
Vr2 = k3./r2;

% Total Potential /Potential Field
V = Vg+Vr1+Vr2;




% Time interval for simulation
tint = [0 14];

% Initial Conditions for x , y & theta
x0 = [0 0 pi/3];

% System dynamics solution using ode23()
[t,Xdot] = ode23('SysDynm',tint,x0);

% Plotting Results (Path)


figure(1)
h = plot(nan,nan);
hold on
grid on
axis([0 14 0 14])
for i = 1:length(Xdot)


    pause(0.1);
    set(h,'XData',Xdot(1:i,1),'YData',Xdot(1:i,2));
    drawnow

end
hold off

figure(2)
meshc(x,y,V)
hold on
plot(Xdot(:,1),Xdot(:,2))
hold off
title('Path of robot')
xlabel('X-axis')
ylabel('Y-axis')
```

Function to solve ODE

-------------SysDynm.m---------------------

```
function Xdot = SysDynm(tint,x)
    % States of the system
     %x = x(1);
     %y = x(2);
     theta = x(3);
    % Gains for the Target and obstacles
k1 =2;
k2 =6;
k3 = 7;


    % Vector to target
    r = sqrt((11-x(1))^2 + (12-x(2))^2);

    % Force on the robot due to target (x-direction)
    Fxt = (k1*(x(1) - 11))/(r);
    % Force on the robot due to taget (y-direction)
    Fyt = (k1*(x(2) - 12))/(r);

    % Vector to obstacle 1 (3,4)
    ro1 = sqrt((3-x(1))^2 + (4-x(2))^2);

    % Force on the robot due to obstacle 1 (3,4)(x-direction)
    Fxo1 = -k2*(x(1) - 3)/((ro1)^(3));
    % Force on the robot due to obstacle 1 (3,4)(x-direction)
    Fyo1 = -k2*(x(2) - 4)/((ro1)^(3));

    % Vector to obstacle 2 (9,7)
    ro2 = sqrt((9-x(1))^2 + (7-x(2))^2);

    % Force on the robot due to obstacle 2 (9,7)(y-direction)
    Fxo2 = -k3*(x(1) - 9)/((ro2)^(3));
    % Force on the robot due to obstacle 2 (9,7)(y-direction)
    Fyo2 = -k3*(x(2) - 7)/((ro2)^(3));

    % Total Forces on the robot (x & y direction)
    Fx = Fxt+Fxo1+Fxo2;
    Fy = Fyt+Fyo1+Fyo2;

    %   Calculating alpha (Theta desired)
    alpha = atan2(Fy,Fx);

    % Calculating phi using the proportional controller
    phi = 4*(alpha - x(3));

    %Assuming constant velocity of the robot to be 3.0

    % System Dynamics passed to ode solver
    Xdot =
[(3*cos(phi)*cos(theta));(3*cos(phi)*sin(theta));(3*sin(phi)/4)];
```

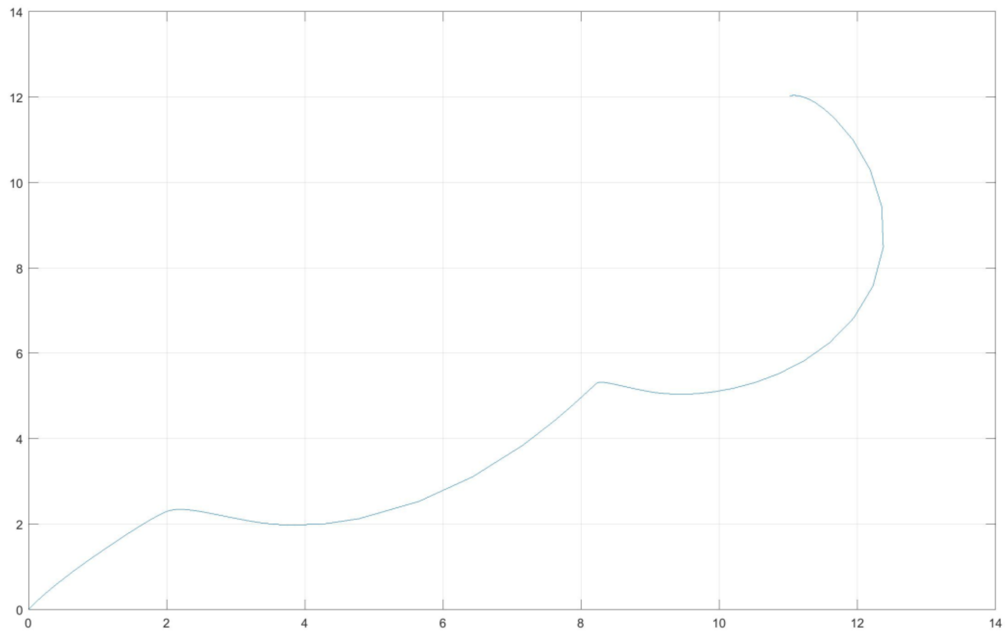


Fig E5 – Path of the robot

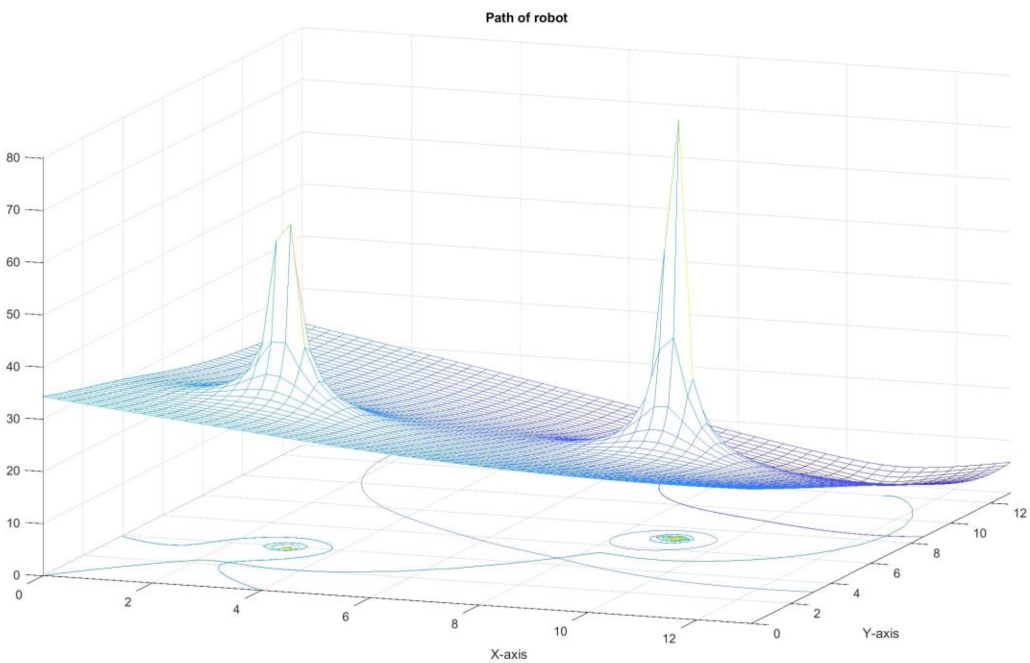
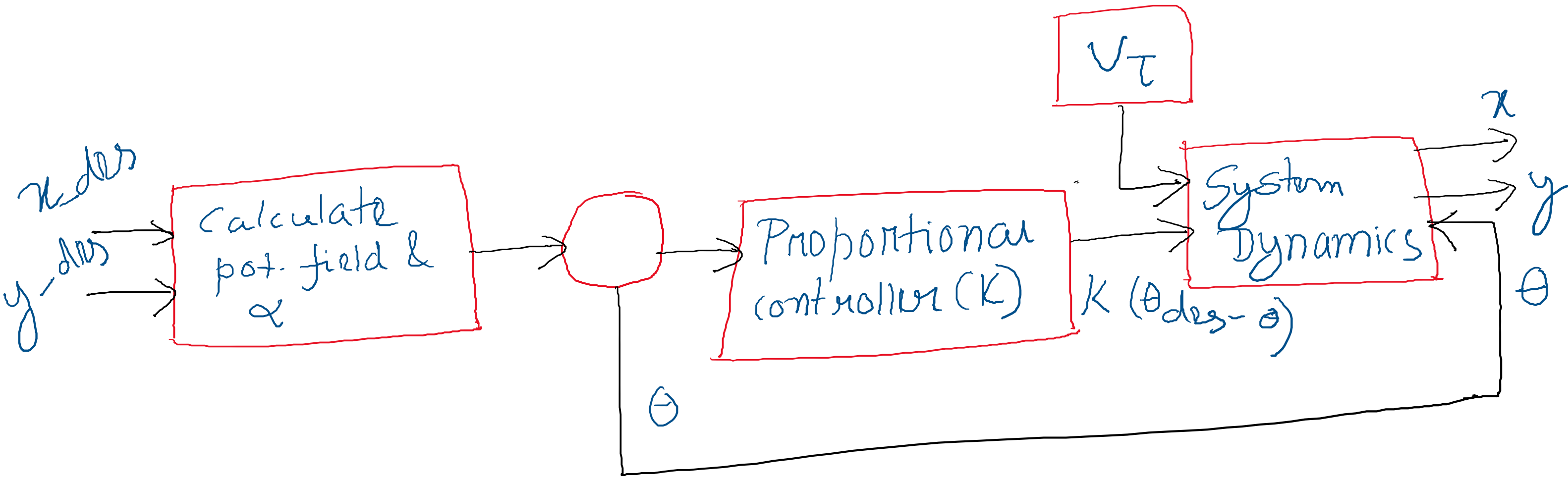


Fig E6 – Path of robot with obstacles

Block diagram of controller



Problem 3 :

Although I didn't get the proper path of the followers, I am still uploading my code. I couldn't get the particular paths due to lack of time. I hope you will consider my efforts for the same.

MATLAB Code :-

```
t=0:0.1:25;
x0=[1 ;1 ;0 ;0; pi/3];
global k
k=[7 4 6];
[t,Xdot] = ode23('followbots',t,x0);

[x,y] = meshgrid(0:0.26:13);
k= 7;


% Potential Field Calculation
r = sqrt((11-x).^2 + (12-y).^2);
Vg = 100*r;
r1 = sqrt((3-x).^2 + (4-y).^2);
Vr1 = 75./r1;
r2 = sqrt((9-x).^2 + (7-y).^2);
Vr2 = 75./r2;
V = Vg+Vr1+Vr2;

% Plotting Leader's path
```

```
figure(1)
h = plot(nan,nan);

grid on
axis([0 14 0 14])
for i = 1:length(Xdot)


    pause(0.001);
    hold on
    set(h,'XData',Xdot(1:i,3),'YData',Xdot(1:i,4));

    drawnow
    hold off

end

figure(2)
  plot(Xdot(:,1),Xdot(:,2))
  hold on
  contour(x,y,V)
  hold off
```

Function Followbots

(Considering only one follower)

```
function Xdot = followbots(t,x)

    % States of the system

    %theta = x(3);
    % Gains for the Target and obstacles

     global k;

    % distance from goal
    rg = sqrt((11-x(1))^2 + (12-x(2))^2);
    % distance from obstacle 1 (3,4)
    r1 = sqrt((3-x(1))^2 + (4-x(2))^2);
    % Distance from obstacle 2 (9,7)
    ro1 = sqrt((x(3)-x(1))^2 + (x(4)-x(2))^2);


    % x-direction Force on the robot due to goal
    Fxg = -k(1)*(x(1) - 11)/(rg);
    % x-direction Force on the robot due to obstacle 1
    Fx1 = k(2)*(x(1) - 3)/((r1)^(3));

    Fxf = k(2)*(x(1) - x(3))/((ro1)^(3));
```

```
 % Total x direction Force on the robot
 Fx = Fxg+Fx1+Fxf;



 % y-direction Force on the robot due to goal
 Fyg = -k(1)*(x(2) - 12)/((rg));
 % x-direction Force on the robot due to obstacle 1
 Fy1 = k(2)*(x(2) - 4)/((r1)^(3));

 Fyf = k(2)*(x(2) - x(3))/((ro1)^(3));

 % Total y direction Force on the robot
 Fy = Fyg+Fy1+Fyf;


%force equation for follower robots


     % distance from goal
 rfg = sqrt(x(1)-x(3))^2 + (x(2)-x(4))^2;
 % distance from obstacle 1 (3,4)
 rf1 = sqrt((3-x(3))^2 + (4-x(4))^2);
 % Distance from obstacle 2 (9,7)
 rf2 = sqrt((9-x(3))^2 + (7-x(4))^2);


 % x-direction Force on the robot due to goal
 Fxfg = -17*(x(3) - x(1))/(rfg);
 % x-direction Force on the robot due to obstacle 1
 Fxf1 = k(2)*(x(3) - 3)/((rf1)^(3));
 % x-direction Force on the robot due to obstacle 2
 FxfL = k(3)*(x(3) - x(1))/((rf2)^(3));

 % Total x direction Force on the robot
 FxfT = Fxfg+Fxf1+FxfL;



 % y-direction Force on the robot due to goal
 Fyfg = -17*(x(4) - x(2))/((rfg));
 % x-direction Force on the robot due to obstacle 1
 Fyf1 = k(2)*(x(4) - 4)/((rf1)^(3));
 % y-direction Force on the robot due to obstacle 2
 FyfL = k(2)*(x(4) - x(2))/((rf2)^(3));

 FyfT=Fyfg+Fyf1+FyfL;


 %   Calculating alpha (Theta desired)
 %alpha = atan2(Fy,Fx);
```

```
    % Calculating phi using the proportional controller
    %phi = 6*(alpha - theta);

    % System Dynamics passed to ode solver
    Xdot = [Fx;Fy;FxfT;FyfT];

end
```